AMENDMENTS TO THE CLAIMS:

This listing of claims will replace all prior versions, and listings, of claims in the application:

LISTING OF CLAIMS:

1. (Currently Amended) A method for managing information exchanges among communicating objects in an object-oriented client server system, said system including first and second object-oriented virtual machines running on counterpart first and second computers in respective server and client roles, and a communication path connection between said computers, said server virtual machine having a run-time environment, the method comprising:

(a) generating a local object at the client machine operable as a proxy to a remote object resident at the server machine; said server machine residing in a smart device; and said client machine having access to the smart device via a smart device reader;

(b) referencing the local object by an application executing at the client machine and causing the local object to marshal parameters; ~~and send~~

(c) sending a process level call request by direct method invocation to the run-time environment of the server machine;

[[(c)**]]** (d) responsive to receipt of said request by the server machine's run-time environment, said run-time environment causing the parameters in the request to become unmarshaled, said remote object to be executed, and ~~replying by marshaling~~ the results of the execution**[[**, and**]]** to be marshaled;

(e) sending a process level return to the client machine as a reply; and

[[(d)**]]** (f) responsive to said reply ~~by the local object operable as a proxy~~, unmarshaling the results from said reply by the local object at the client machine.

2. (Currently Amended) The method according to claim 1, wherein **[[**said**]]** plural process call level requests and ~~said reply~~ replies are generated in an alternating manner.

3. (Original) The method according to claim 1, wherein the local object when operating as a proxy at the client machine and the run-time environment when operating at the server machine perform respectively as stubs.

4. (Currently Amended) A method for managing information exchanges between an application executing at a object-oriented virtual machine operable as a client and a remote object resident at another object-oriented virtual machine operable as a server, said server machine having a run-time environment, said client and server having a communication path connection there-between, said communication path connection being operable under a process for originating and sending byte level messages therebetween, comprising:

(a) providing a local object resident at the client machine operable as a proxy stub to the remote object resident at the server machine and providing a description of the remote object to enable said run-time environment to also operate as a stub, said server machine residing in a smart device; and said client machine having access to the smart device via a smart device reader;

(b) responsive to a client application call to the local object, marshaling parameters and causing a process level call request to be sent to the ~~remote object at~~ run-time environment of the server machine, said sending of the request further including mapping said process level call request into counterpart byte string level messages and transmitting said messages to the server machine;

(c) responsive to receipt of said request messages by the server machine's run-time environment, mapping said messages into a process level call request, unmarshaling the parameters, invoking and executing the remote object, marshaling the results, forming a process level reply, mapping said reply into string byte messages, and transmitting said reply messages to the client machine; and

(d) responsive to the reply messages by the proxy at the client machine, mapping said reply messages into a process level reply, and unmarshaling the results.

5. (Original) The method according to claim 4, wherein said object-oriented virtual machines include Java virtual machines, and further wherein the remote object is an applet, and the local object is an interface description.

6. (Currently Amended) An article of manufacture comprising a machine readable memory having stored therein a plurality of processor executable control program steps for managing information exchanges among communicating objects in an ~~objects~~ object-oriented client server system, said system including first and second object-oriented virtual machines running on counterpart first and second computers in respective server and client roles, and a communication path

connection between said computers, said server virtual machine having a run-time environment, said control program steps ~~include~~ including:

(a) a control program step for generating a local object at the client machine operable as a proxy to a remote object resident at the server machine, said server machine residing in a smart device; and said client machine having access to the smart device via a smart device reader;

(b) a control program step for referencing the local object by an application executing at the client machine and causing the local object to marshal parameters;

(c) a control program step for transmitting ~~and send~~ a process level call request to the server ~~machine~~ machine's run-time environment;

[[(c)**]]** (d) a control program step **[[**for**]]** responsive to receipt of said request by the server machine's run-time environment, to cause said run-time environment ~~causing~~ to unmarshal the parameters in the request ~~to become unmarshaled~~, execute said remote object ~~to be executed~~, marshal the results of the execution ~~marshaled~~, and send a process level return **[[**sent**]]** to the client machine; and

[[(d)**]]** (e) a control program step **[[**for**]]** responsive to said ~~reply by the~~ return to cause said local object ~~operable as a proxy, unmarshaling~~ to unmarshal the results from said reply.

7. (Previously Presented) The method according to claim 1, wherein said client machine accesses the smart device with communication protocols specified according to International Standards Organization specification 7816-4.

8. (Previously Presented) The method according to claim 7, wherein said client machine obtains access to the smart device via a command Application Program Data Unit.

9. (Previously Presented) The method according to claim 1, wherein said reply is formatted into an Application Program Data Unit response.

10-13. (Canceled).

14. (Currently Amended) An object-oriented system comprising:

a client computer comprising:

an application configured to generate a local call on the client computer to invoke a method of an applet; and

an applet proxy configured to generate a single command APDU in response to the local call on the client computer, the applet proxy being a local object to the application; and

a smart device comprising:

the applet, the applet being a remote object to the application and configured to invoke the method in response to a local call on the smart device; and

a run-time **[[**time**]]** environment configured to generate the local call on the smart device to invoke the method in response to the single command APDU without the applet having been selected with another command APDU.

15. (Previously Presented) The object-oriented system of claim 14 wherein:

the applet proxy is further configured to marshal parameter values from the local call on the client computer in the single command APDU; and

the run-time environment is further configured to unmarshal the parameter values in the single command APDU and provide them to the method with the local call on the smart device.

16. (Previously Presented) The object-oriented system of claim 14 wherein:

the method is configured to generate a local return on the smart device to return to the application after the method is completed;

the run-time environment is further configured to generate a single response APDU in response to the local return on the smart device; and

the applet proxy is further configured to generate a local return on the client computer to return to the application in response to the single response APDU.

17. (Previously Presented) The object-oriented system of claim 16 wherein:

the applet proxy is further configured to marshal parameter values from the local call on the client computer in the single command APDU;

the run-time environment is further configured to unmarshal the parameter values in the single command APDU and provide them to the method with the local call on the smart device;

the run-time environment is further configured to marshal return values from the local return on the smart device in the single response APDU; and

the applet proxy is further configured to unmarshal the return values in the single response APDU and provide them to the application in the local return on the client computer.

18. (Previously Presented) The object-oriented system of claim 17 wherein the application, the applet proxy, and the applet are written in Java and the run-time environment is a Java card run-time environment.

19. (Previously Presented) A process of making a direct method invocation in an object-oriented system, the object-oriented system including a client computer and a smart device, the client computer having an application and an applet proxy, the applet proxy being a local object to the application, the smart device having a run-time environment and an applet with a method, the applet being a remote object to the application, the process comprising the steps of:

with the application, generating a local call on the client computer to invoke the method;

with the applet proxy, generating a single command APDU in response to the local call on the client computer;

with the run-time environment, generating a local call on the smart device to invoke the method in response to the single command APDU without the applet having been selected with another command APDU; and

with the applet, invoking the method in response to the local call on the smart device.

20. (Previously Presented) The method of claim 19 further comprising the steps of:

with the applet proxy, marshalling parameter values from the local call on the client computer in the single command APDU; and

with the run-time environment, unmarshalling the parameter values in the single command APDU and providing them to the method in the local call on the smart device.

21. (Previously Presented) The method of claim 19 wherein:

with the method, generating a local return on the smart device to return to the application after the method is completed;

with the run-time environment, generating a single response APDU in response to the local return on the smart device; and

with the applet proxy, generating a local return on the client computer to return to the application in response to the single response APDU.

22. (Previously Presented) The method of claim 21 further comprising the steps of:

with the applet proxy, marshalling parameter values from the local call on the client computer in the single command APDU;

with the run-time environment, unmarshalling the parameter values in the single command APDU and providing them to the method in the local call on the smart device;

with the run-time environment, marshalling return values from the local return on the smart device in the single response APDU; and

with the applet proxy, unmarshalling the return values in the single response APDU and providing them to the application in the local return on the client computer.

23. (Previously Presented) The method of claim 22 wherein the application, the applet proxy, and the applet are written in Java and the run-time environment is a Java card run-time environment.

24. (Previously Presented) A smart device for use with a client computer in an object-oriented system, the client computer comprising an application and an applet proxy, the applet proxy being a local object to the application, the smart device comprising:

a run-time time environment configured to generate a local call on the smart device to invoke a method of an applet in response to a single command APDU without the applet having been selected with another command APDU, the single command APDU being previously generated by the applet proxy in response to a local call by the application on the client computer to invoke the method; and

the applet, the applet being a remote object to the application and configured to invoke the method in response to the local call on the smart device.

25. (Previously Presented) The smart device of claim 24 wherein the run-time environment is further configured to unmarshal parameter values in the single command APDU and provide them to the method with the local call on the smart device, the parameter values being previously marshaled by the applet proxy in the single command APDU after being provided to the applet proxy with the local call on the client computer.

26. (Previously Presented) The smart device of claim 24 wherein:

the method is configured to generate a local return on the smart device to return to the application after the method is completed;

the run-time environment is further configured to generate a single response APDU in response to the local return on the smart device, a local return on the client computer being subsequently generated by the applet proxy to return to the application in response to the single response APDU.

27. (Previously Presented) The smart device of claim 26 wherein:

the run-time environment is further configured to unmarshal parameter values in the single command APDU and provide them to the method with the local call on the smart device, the parameter values being previously marshaled by the applet proxy in the single command APDU after being provided to the applet proxy with the local call on the client computer

the run-time environment is further configured to marshal return values from the local return on the smart device in the single response APDU, the return values

in the single response APDU being subsequently unmarshalled and provided to the application with the local return on the client computer by the applet proxy.

28. (Previously Presented) The smart device of claim 24 wherein the application, the applet proxy, and the applet are written in Java and the run-time environment is a Java card run-time environment.

29. (Previously Presented) The method of claim 1, wherein the smart device comprises a smart card.

30. (Previously Presented) The method of claim 4, wherein the smart device comprises a smart card.

31. (Previously Presented) The article of manufacture of claim 6, wherein the smart device comprises a smart card.